Software Architecture Fundamentals: Technical, Business, and Social Influences

Rob Wojcik
Software Engineering Institute

Webinar
July 8th, 2010

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **08 JUL 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **Software Architecture Fundamentals: Technical, Business, and Social Influences**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**34**	

Today's Speaker

Rob Wojcik is a member of the technical staff in the Research, Technology, and System Solutions Program at the SEI, a position he has held since 2004. In his current position, he performs training and consulting in software architecture technology and software architecture evaluations. Prior to his position at the SEI, over twenty five years of his career were dedicated to developing software systems that utilize traditional as well as artificial intelligence and object oriented technologies.

Polling Question #1

How did you hear about this webinar?

1. Email invitation from the SEI
2. SEI Website
3. Website with webinar calendar (i.e., www.webinar-directory.com)
4. Social Media (i.e., LinkedIn, Twitter)
5. SEI Member Bulletin

Software Engineering Institute (SEI)

Department of Defense R&D Laboratory (FFRDC)

Created in 1984

Under contract to Carnegie Mellon University

Offices in Pittsburgh, PA; Washington, DC; and Frankfurt, Germany

SEI Mission: advance software and related disciplines to ensure the development and operation of systems with predictable and improved cost, schedule, and quality.

Research, Technology, and System Solutions (RTSS)Program

Vision

- Enable assured and flexible system capabilities at all scales.

Mission

- Focus on the structure and behavior of software-reliant systems

Portfolio of RTSS Program Work

Initiatives

- Architecture-Centric Engineering
- System of Systems Practice
- System of Systems Software Assurance
- Product Line Practice

Cross-Cutting Efforts

- Concept Lab
- Integrating Solutions
- Ultra-Large-Scale Systems

Independent Research and Development

Today’s Topics

- What is Software Architecture?
- Why is Software Architecture Important?
- Which Requirements Are Most Important To Architectural Design?
- What Else Influences Software Architecture?

I’ll take questions at the end of the presentation.

How Much Do You Know?

a) I know a whole lot about software architecture

b) I know enough about software architecture to get by

c) I know very little about software architecture

d) What the heck is software architecture?

Today’s Topics

- **What is Software Architecture?**
- Why is Software Architecture Important?
- Which Requirements Are Most Important To Architectural Design?
- What Else Influences Software Architecture?

Many Definitions for Software Architecture

Alfred: Software architecture consists of the rules and principles for how a system is decomposed into its component parts, the rationale for how responsibilities are allocated among those parts, and the policies and mechanisms that coordinate the interactions between those parts as they collaborate to fulfill the purpose of the system. Software architecture is at once the partitioning of a system into its significant elements, and the organization and integration of those elements into a cohesive whole.

RUP: the set of significant decisions about the organization of a software system, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style that guides this organization---these elements and their interfaces, their collaborations, and their composition

Pandey: the blueprint of the various framework components that coordinate together to satisfy the design guidelines of a specific domain

Ramanujam: an iterative framework between software components required to meet the stated objectives of the business, in terms of cost to develop/maintain the software components, time to market, and life expectancy of the components

Software Engineering Institute: the structure or structures of the system, which comprise the software elements, the externally visible properties of those elements, and the relationships among them

ANSI/IEEE: the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution

Adabala: a style that is proven scientifically and adopted by the engineering discipline, with which a software is developed so as to sustain and adopt to the growing needs of the industry from time to time...

Riemenschneider: the mapping from the problem space to the solution space, where the problem space is devoid of implementation concerns and the solution space is the sum total of all implementation concerns

Mulvaney: a set of implementation elements together with the mechanisms through which they collaborate to provide the system's required functionality

Ahmed: a coherent set of abstract patterns, or principles, guiding the design of each aspect of a large software system…

Matthaeus: A configurable skeleton of any kind of software beast on which you hang implementation specific muscle to make it live

Some Things Remain Certain - 1

Software architecture

- is an abstraction that describes software elements
- addresses the roles, responsibilities, behaviors and properties of software elements
- addresses the relationships between software elements
- shows what software elements provide to and require from each other
- shows the relationship to non-software elements
- is described from many different perspectives

Some Things Remain Certain - 2

Every software system has an architecture.

A software architecture is not inherently good or bad.

Today’s Topics

- What is Software Architecture?
- **Why is Software Architecture Important?**
- Which Requirements Are Most Important To Architectural Design?
- What Else Influences Software Architecture?

Why Is Software Architecture Important?

It’s a vehicle for communication.

It’s a manifestation of earliest design decisions that

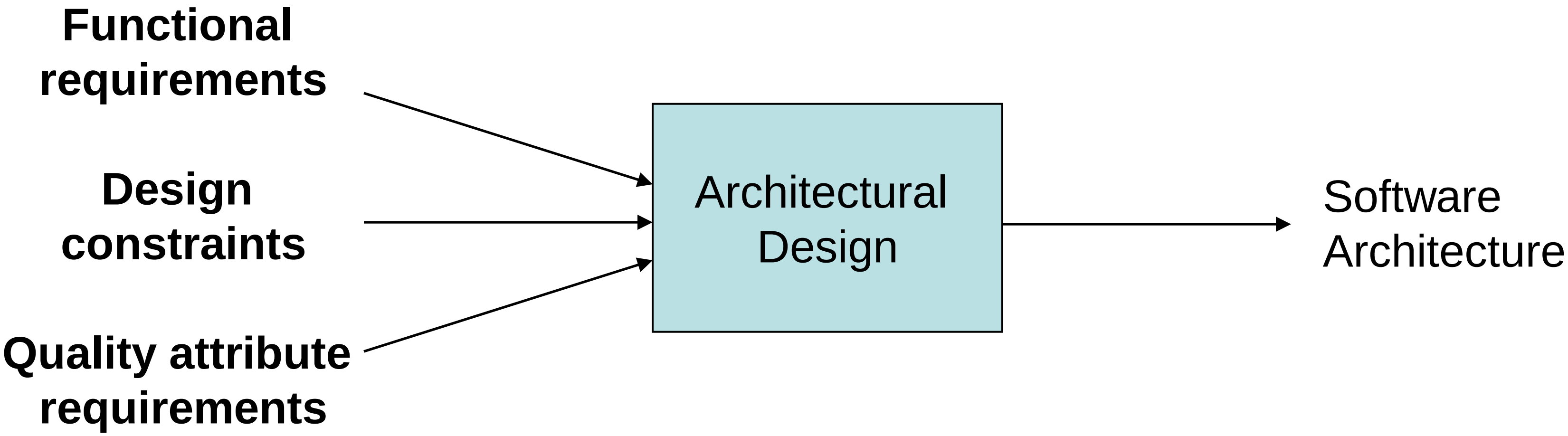
- defines implementation constraints
- relates to organizational structure
- provides the basis for project artifacts and activities
- permits/precludes achieving requirements
- allows us to predict system qualities
- allows us to control complexity
- allows us to reason about and manage change
- allows us to develop a skeletal system
- provides a sound basis for cost and schedule estimates

It’s a transferable, reusable abstraction.

Today’s Topics

- What is Software Architecture?
- Why is Software Architecture Important?
- **Which Requirements Are Most Important To Architectural Design?**
- What Else Influences Software Architecture?

Which Requirements Are Most Important to Architectural Design?



What determines whether these requirements are met?

Which requirements are the most important when it comes to structuring an architecture?

Something to Consider

What’s wrong with designing a system that has one big source module, one big object module, and one big executable as long as it functions properly?

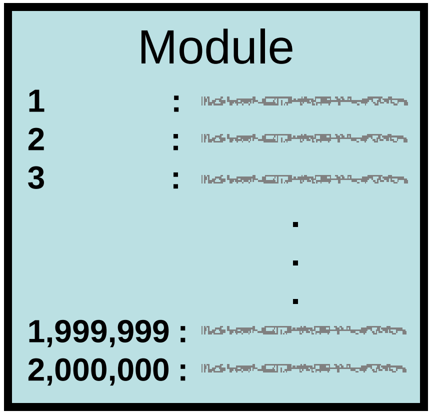
buildability

modifiability

testability

complexity

maintainability


Module

1 :

2 :

3 :

.

.

.

1,999,999 :

2,000,000 :

portability

reliability

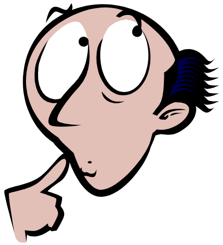
distributability

availability

reusability

Others?

Which requirements do you think would be negatively impacted by this “design”?



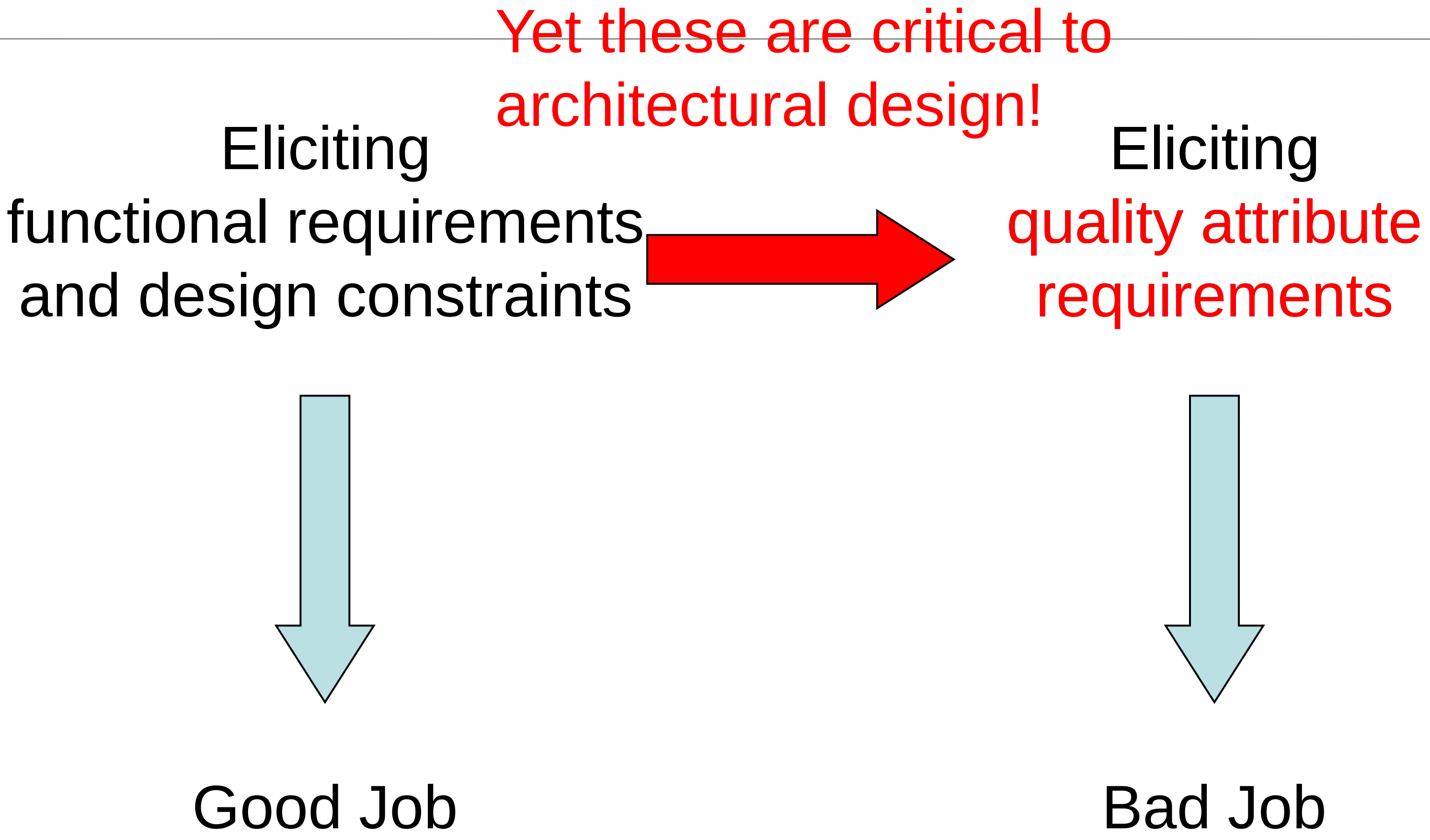
Here’s the Point!

If functionality is the only thing that matters, any software architecture will do!

It’s the requirements that are above and beyond functionality that require us to structure an architecture. They include:

- design constraints
- quality attributes

Requirements Elicitation



Yet these are critical to architectural design!

Eliciting functional requirements and design constraints

Eliciting quality attribute requirements

Good Job

Bad Job

Difficulties in Eliciting Quality Attribute Requirements

Non-Operational requirements

- “The system must be easy to use.”
- “The system must have high performance.”
- “The system must be portable.”

Debating the quality attribute to which a system behavior belongs

- “The system must process 10,000 messages per second.”

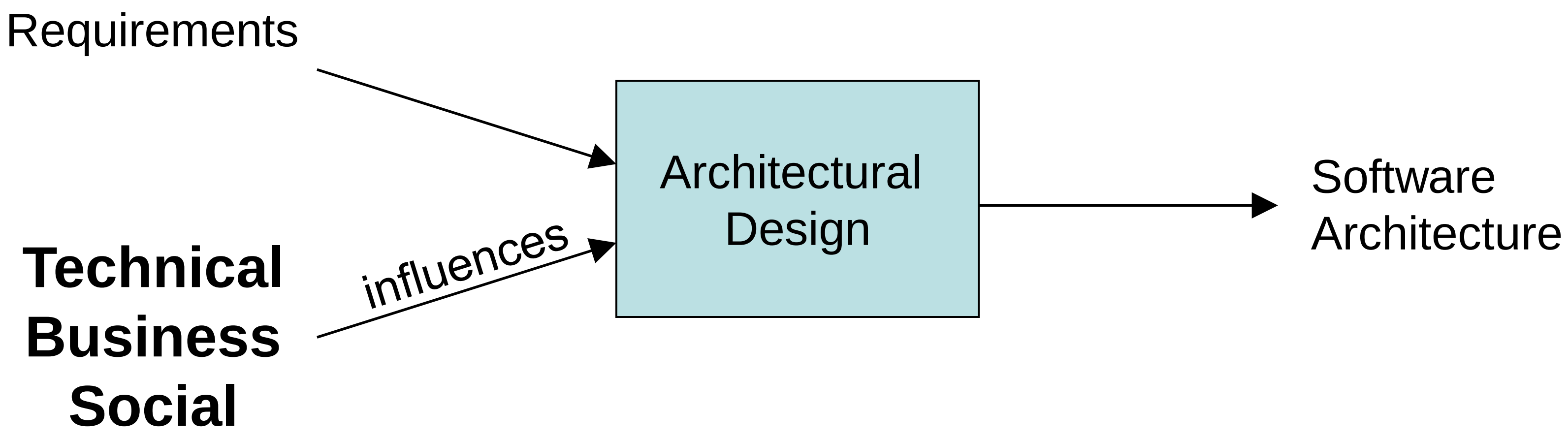
Vocabulary variations

- Everyone knows what “high performance” means, right?

Today's Topics

- What is Software Architecture?
- Why is Software Architecture Important?
- Which Requirements Are Most Important To Architectural Design?
- **What Else Influences Software Architecture?**

Other Influences on the Architecture



Software architecture is influenced by the technical, business, and social environment.

Examples of Other Influences

Stakeholders

- customers, users, managers, marketing, developers, maintainers, etc.

Development organization

- immediate and long term business goals
- organizational structure

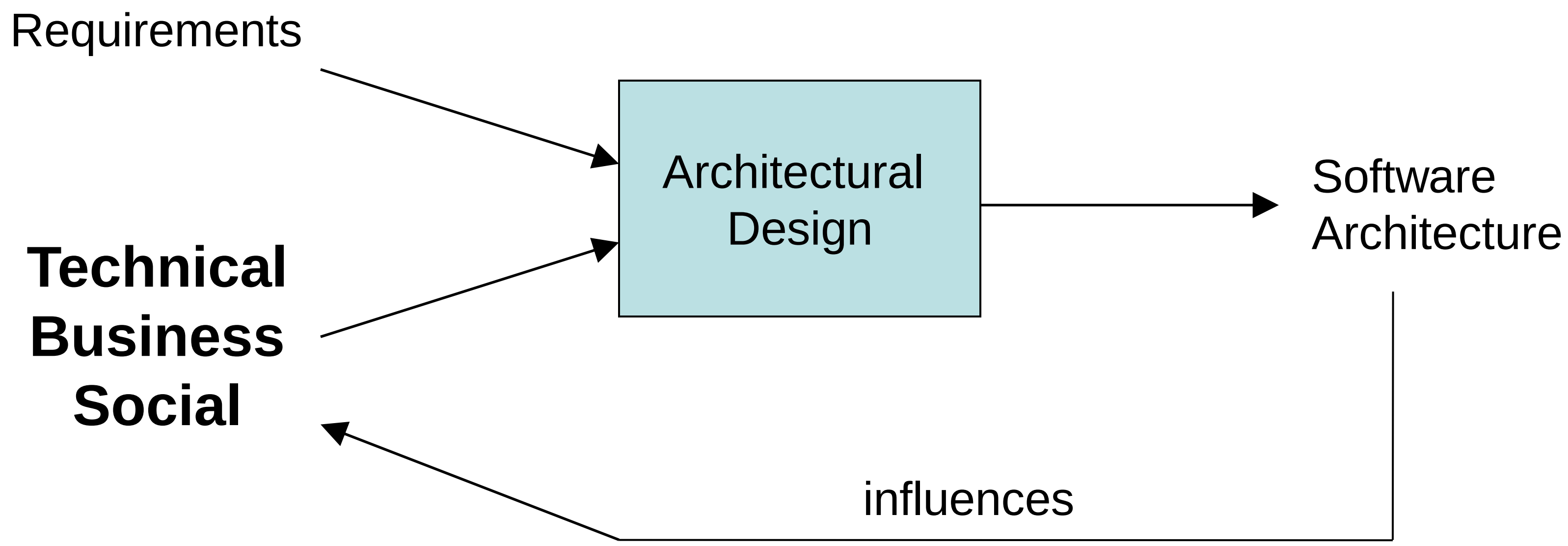
Technical environment

- object oriented, WWW, intelligent agents, EJB, service oriented, J2EE, thin client, .NET, etc.

Background and experience

- architect and organizational experience
- education and training

What Architecture Influences



An architecture can influence the technical, business, and social environment.

Examples of What Architecture Can Influence

Development organization

- structure, goals, artifacts, etc.

Stakeholder requirements

- demand for similar features, existing components and system

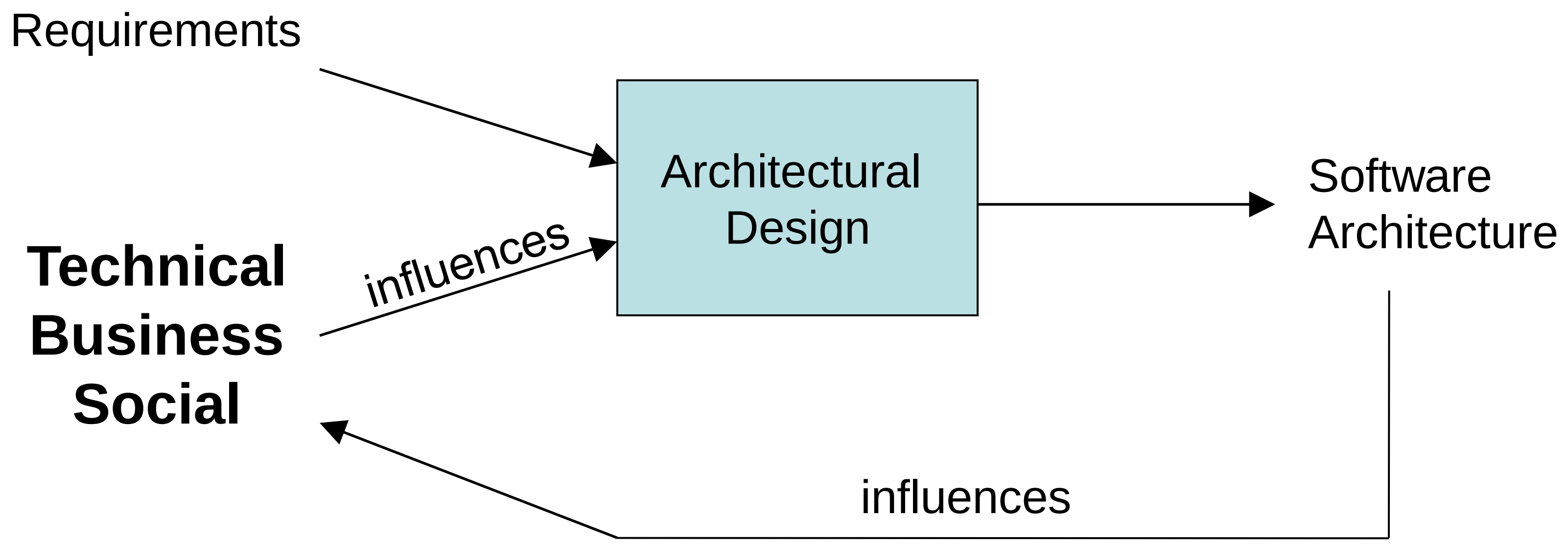
Technical environment

- relational databases, WWW, service oriented architectures, etc.

Background and experience

- promote approaches that have been successful
- reject approaches that have failed

Understanding These Influences



Understanding this cycle of influences helps us to plan for and manage change throughout the lifetime of a system.

Conclusion

Software architecture is important!

Every software system has an architecture!

Quality attribute requirements are critical!

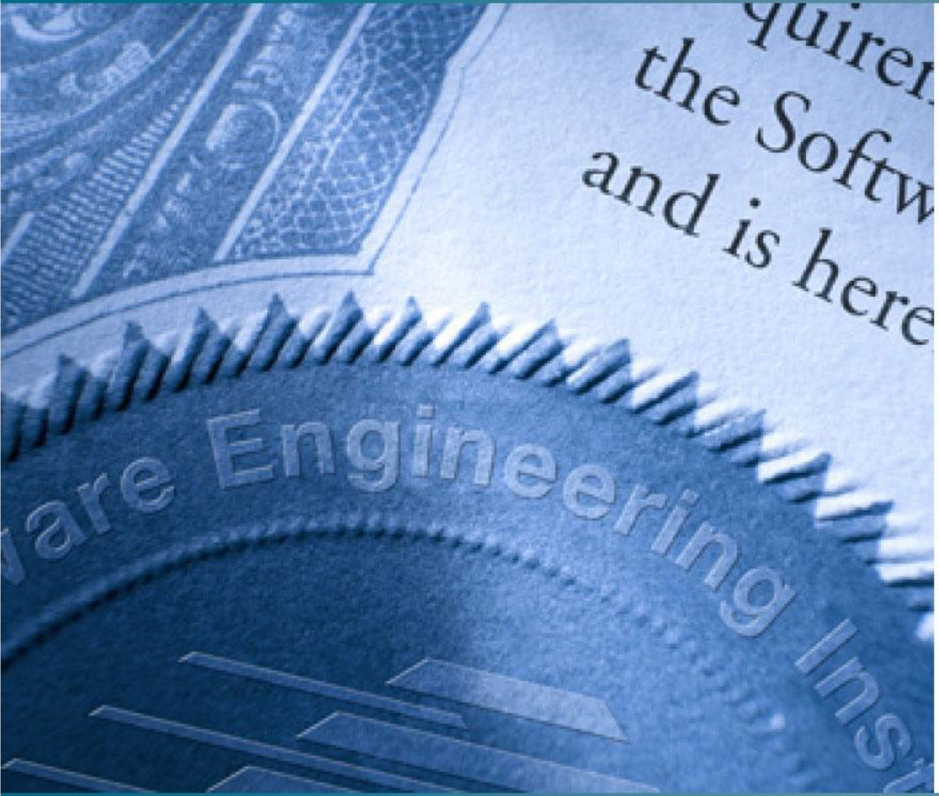
Requirements aren’t the only things that influence software architectures!

For More Information

Software Architecture in Practice, 2nd edition
written by Len Bass, Paul Clements, & Rick Kazman
and published by Addison-Wesley as part of the
SEI Series in Software Engineering

Other information is provided at
http://www.sei.cmu.edu/architecture/.

Contact Rob Wojcik
Software Engineering Institute
rwojcik@sei.cmu.edu



Proof of your skill from a world leader in software engineering.

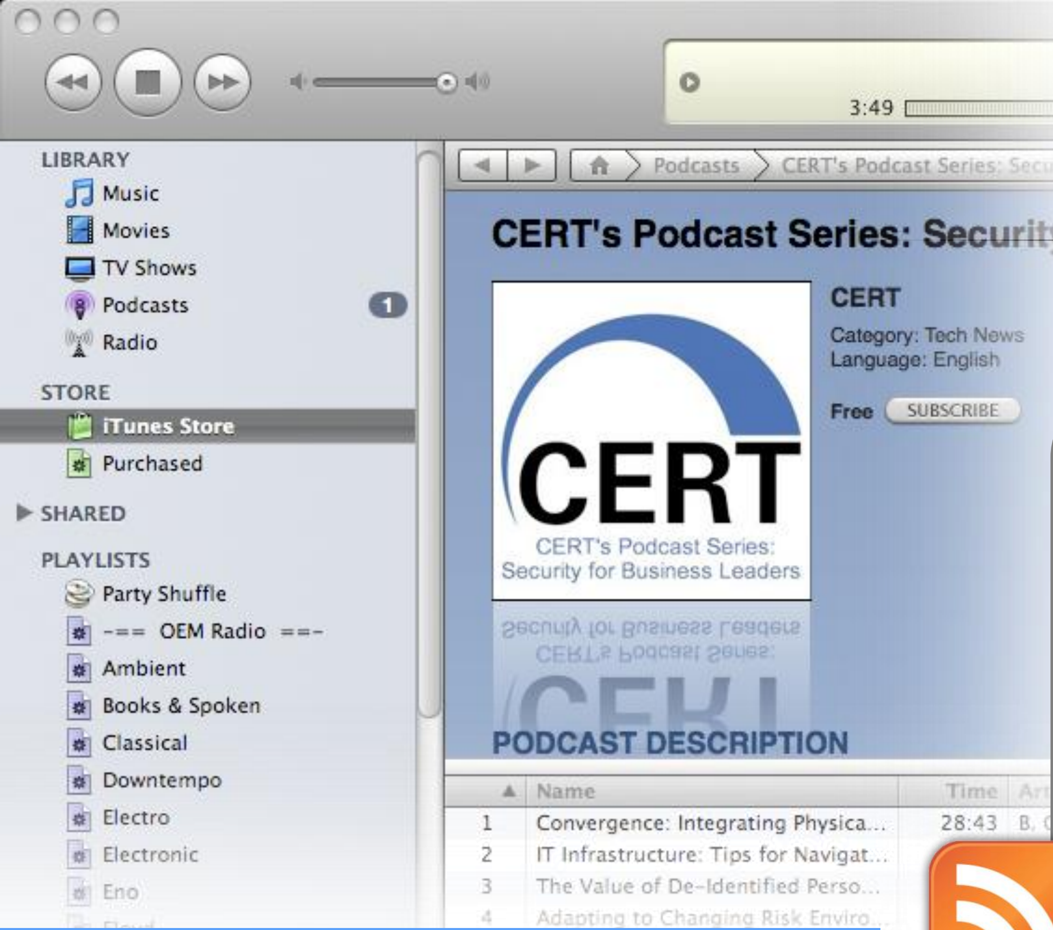
Become an SEI-Certified ATAM Leader.

▶ www.sei.cmu.edu/certification/architecture/atam

Want a Closer Connection to the SEI?

Become an SEI Member!

▶ www.sei.cmu.edu/membership



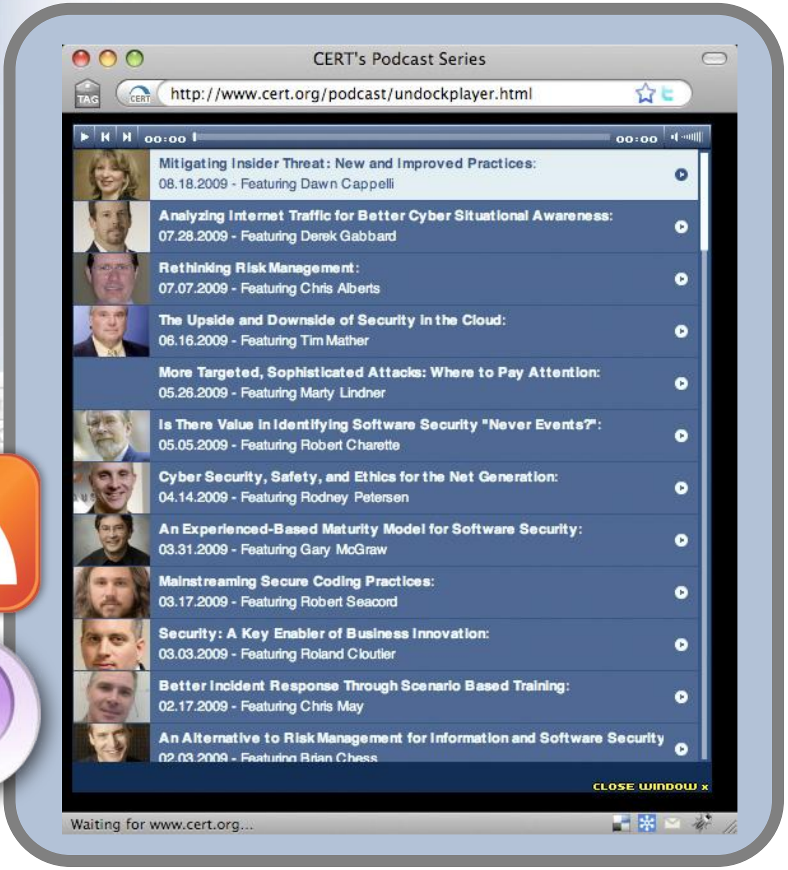
CERT's Podcast Series: Security for B

CERT's Podcast Series: Security for Business Leaders

CERT

Category: Tech News
Language: English

	Name	Time	Art
1	Convergence: Integrating Physica…	28:43	B. (
2	IT Infrastructure: Tips for Navigat…		
3	The Value of De-Identified Perso…		
4	Adapting to Changing Risk Enviro…		



CERT's Podcast Series

http://www.cert.org/podcast/undockplayer.html

Mitigating Insider Threat: New and Improved Practices:
08.18.2009 - Featuring Dawn Cappelli

Analyzing Internet Traffic for Better Cyber Situational Awareness:
07.28.2009 - Featuring Derek Gabbard

Rethinking Risk Management:
07.07.2009 - Featuring Chris Alberts

The Upside and Downside of Security in the Cloud:
06.16.2009 - Featuring Tim Mather

More Targeted, Sophisticated Attacks: Where to Pay Attention:
05.26.2009 - Featuring Marty Lindner

Is There Value in Identifying Software Security "Never Events?":
05.05.2009 - Featuring Robert Charette

Cyber Security, Safety, and Ethics for the Net Generation:
04.14.2009 - Featuring Rodney Petersen

An Experienced-Based Maturity Model for Software Security:
03.31.2009 - Featuring Gary McGraw

Mainstreaming Secure Coding Practices:
03.17.2009 - Featuring Robert Seacord

Security: A Key Enabler of Business Innovation:
03.03.2009 - Featuring Roland Cloutier

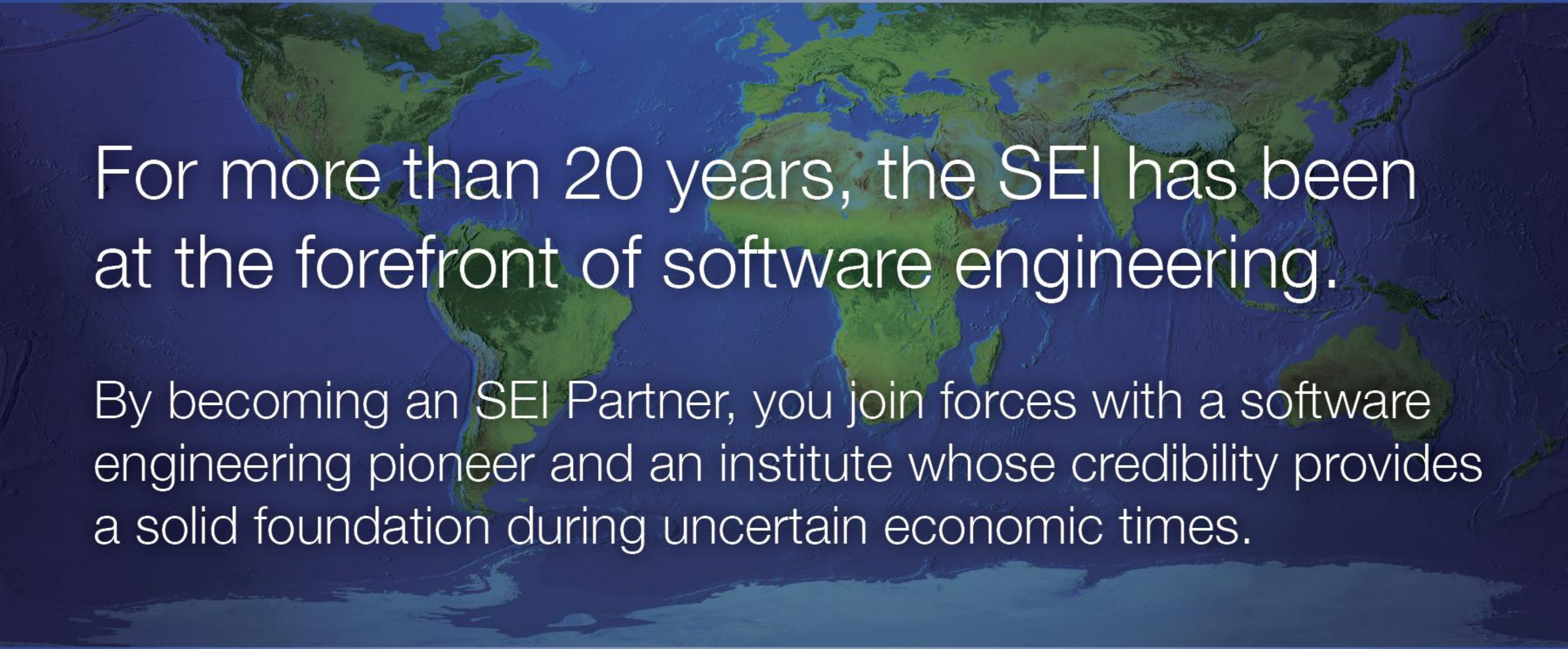
Better Incident Response Through Scenario Based Training:
02.17.2009 - Featuring Chris May

An Alternative to Risk Management for Information and Software Security
02.03.2009 - Featuring Brian Chess

Waiting for www.cert.org...

CERT's Podcast Series:
Security for Business Leaders

▶ www.cert.org/podcast/



For more than 20 years, the SEI has been at the forefront of software engineering.

By becoming an SEI Partner, you join forces with a software engineering pioneer and an institute whose credibility provides a solid foundation during uncertain economic times.

SEI Partner Network

▶ www.sei.cmu.edu/partners

Do you have the knowledge you need?

SEI Training

▶ www.sei.cmu.edu/training

- NO WARRANTY

- THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

- Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

- This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

- This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.